Precision Formation Flying Integrated Analysis Tool (PFFIAT)

Bob Schwenk

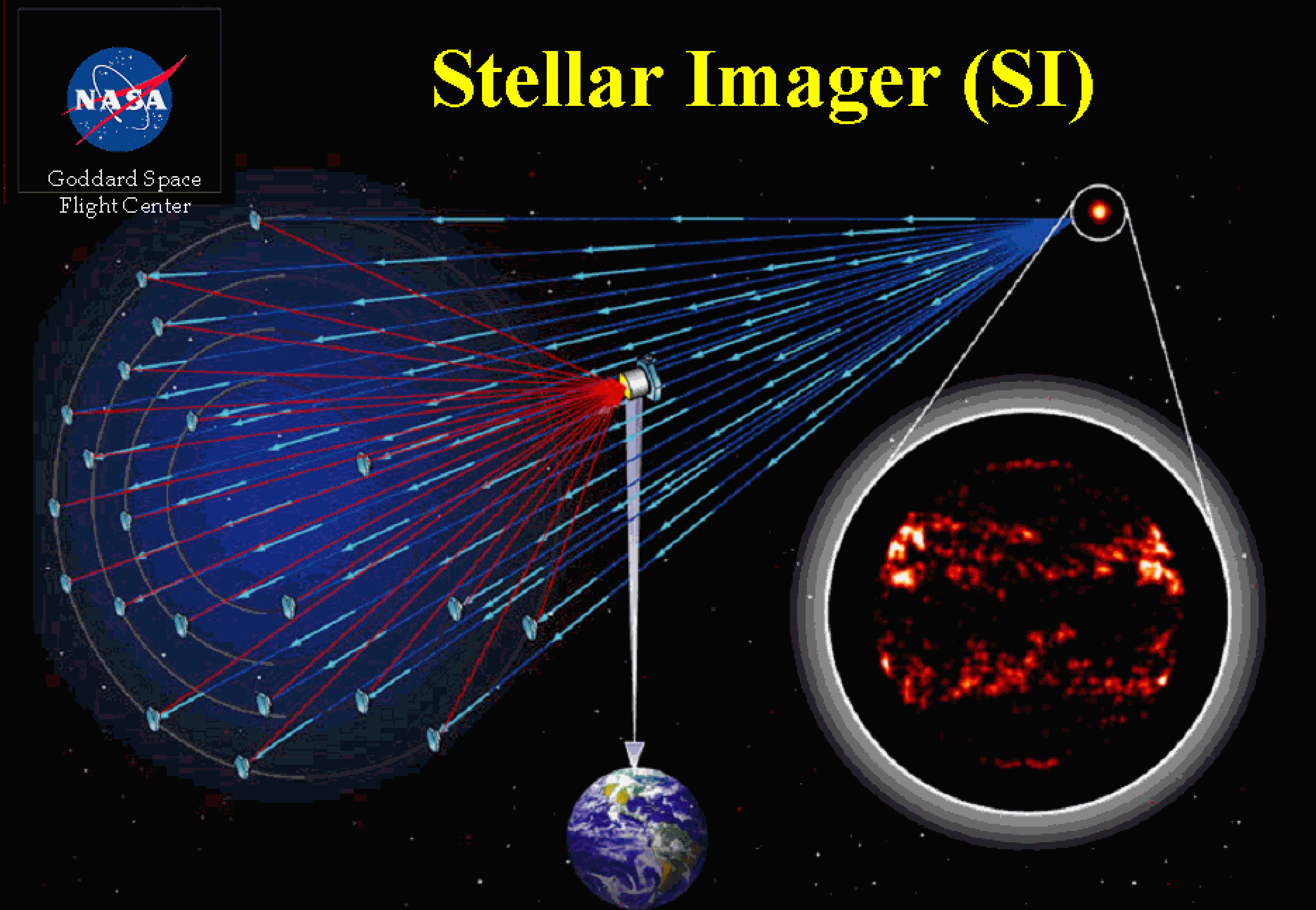
PFFIAT Overview

- Part of Internal Research and Development (IRAD) Program, initiated in Fiscal Year (FY) 2002
- GOAL: Tie together dynamics of satellite formations to interferometric and/or optical performance metrics

Rationale

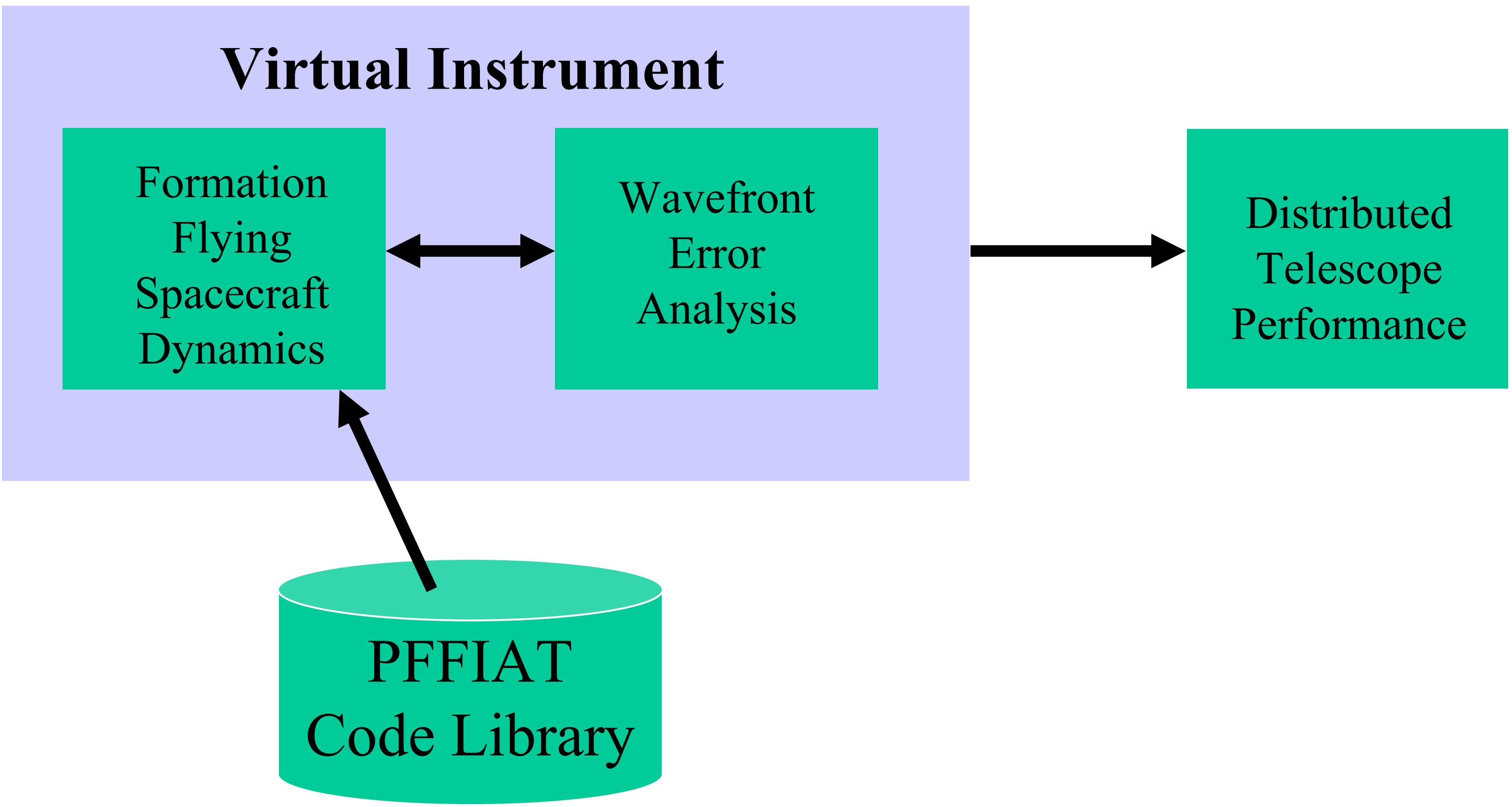
- PFFIAT enables productive and immediate first order assessment of mission concepts
- PFFIAT supports mission design for future Precision Formation Flying missions, including:
  - Stellar Imager (SI)
  - Micro-Arcsecond X-ray Imaging Mission (MAXIM)

Stellar Imager (SI)



http://hires.gsfc.nasa.gov/~si

PFFIAT Concept



PFFIAT Team

- Code 571 – Dr. Jesse Leitner
- Code 572 – Dave Folta, Steven Hughes, Scott Sarin, Steven Cooley
- Code 583 – Bob Schwenk
- Code 600 – Dr. Keith Gendreau

Target Customers and Missions

- Mission Designers
  - Model the virtual instrument
  - Design spacecraft trajectories
  - Evaluate control laws
  - Evaluate fuel usage
- Proof-of-Concept Demonstrations
  - LEONARDO (2008)
  - MAXIM Pathfinder (2012)
  - SPECS, MAXIM, SI (2016)

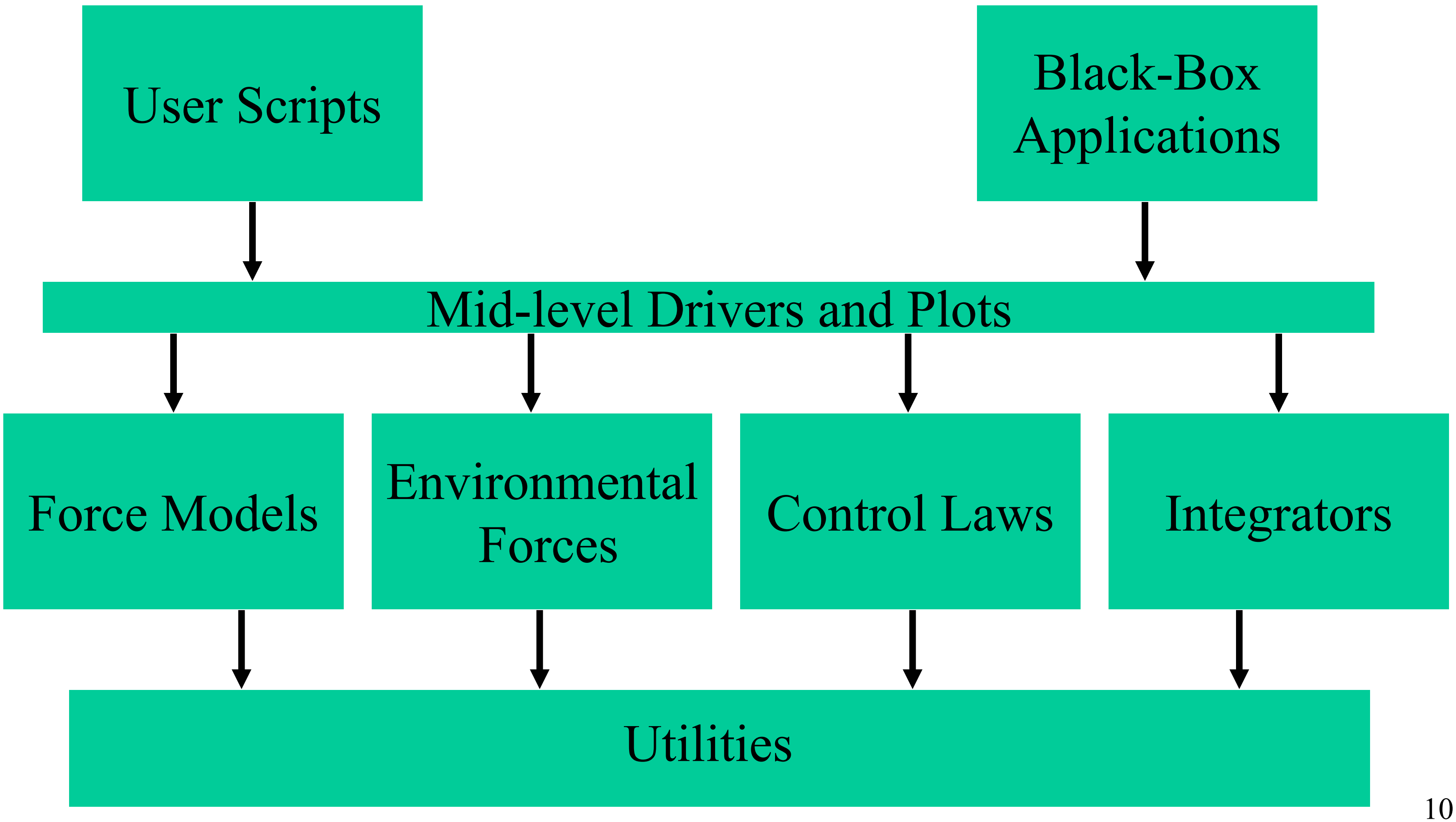
PFFIAT Code Library Requirements

- Quickly generate user-modifiable scripts for use by analysts
- Provide a set of black-box applications
- Read and write standard files
- Provide plots for analysis and presentation
- Use MATLAB, including mex-files
- Read the Solar-Lunar-Planetary (SLP) ephemeris file

PFFIAT Code Library Requirements

- Propagate N spacecraft given initial conditions
- Implement restricted three-body problems
- Allow choice of integrators (e.g., ODE113)
- Allow input and output in various coordinate and time systems
- Allow different integrators and force models for each spacecraft in the formation
- Implement various continuous-thrust control laws

Architecture



Development Environment

- Windows Platform
- MATLAB 6.5 (R13) with C and FORTRAN mex-files
- RCS Configuration Control

Typical Implementation Cycle

- Analyst Provides:
  - New scenario
  - New control law
  - New force models
  - New plot requirements
  - Performance requirements
- Developer Responds:
  - New scenario scripts
  - Implementation and testing of new code
  - New official release delivered to analyst

Cycle requires anywhere from 1 week to 3 months

Near-Term Applications

Types of Missions:

Multiple Spacecraft at a Lagrange Point

Solar Sail-Craft Trajectories

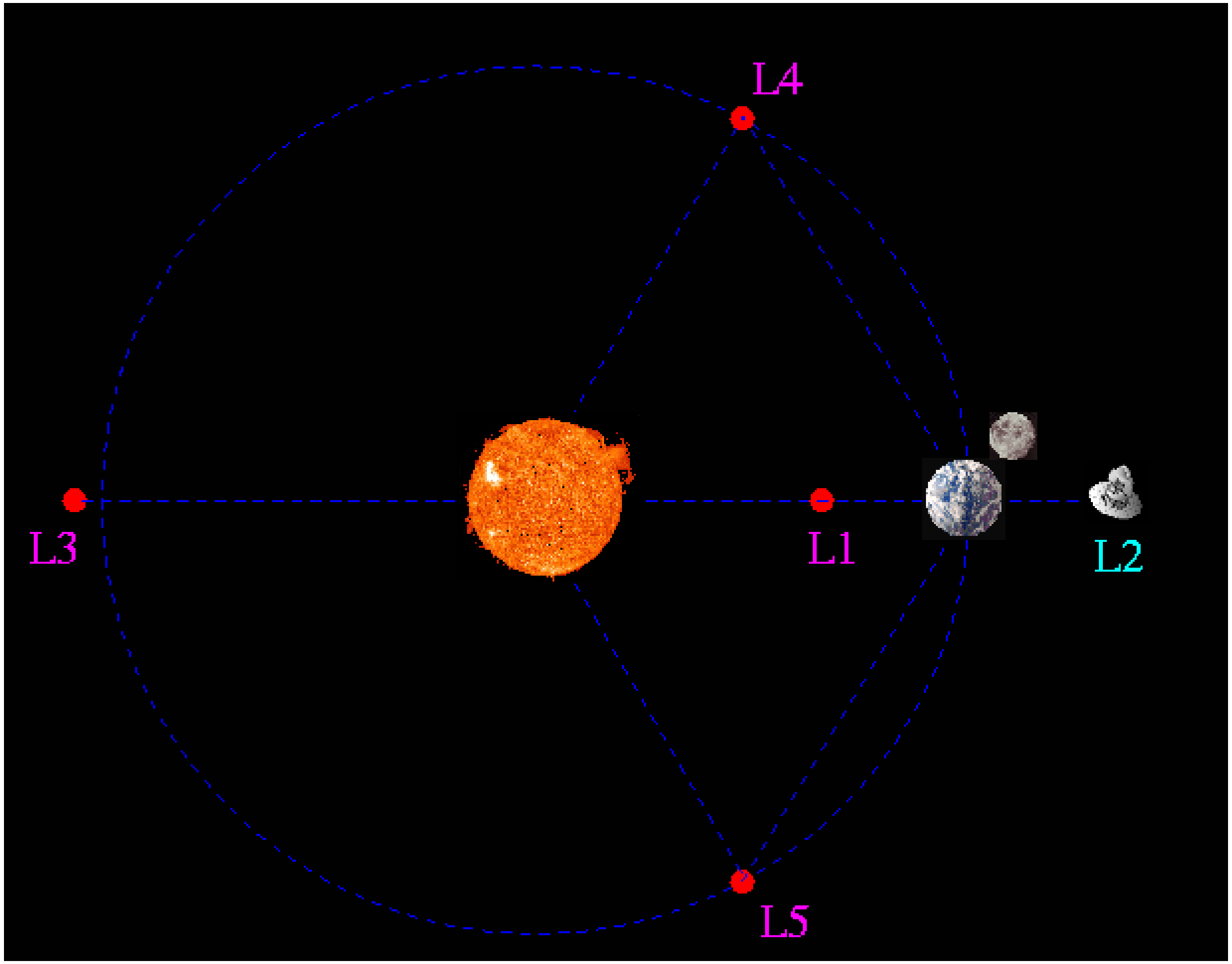
Typical Analyses:

Evaluate Control Laws

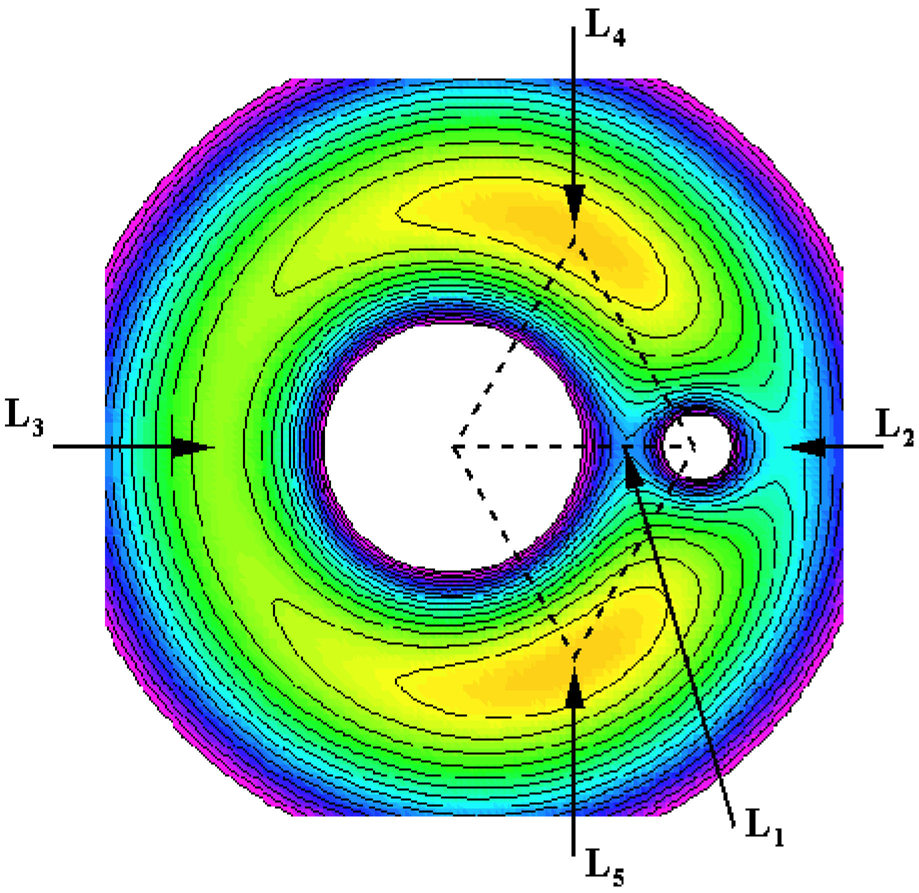
Predict Fuel Usage

Optimize Transfer Trajectory

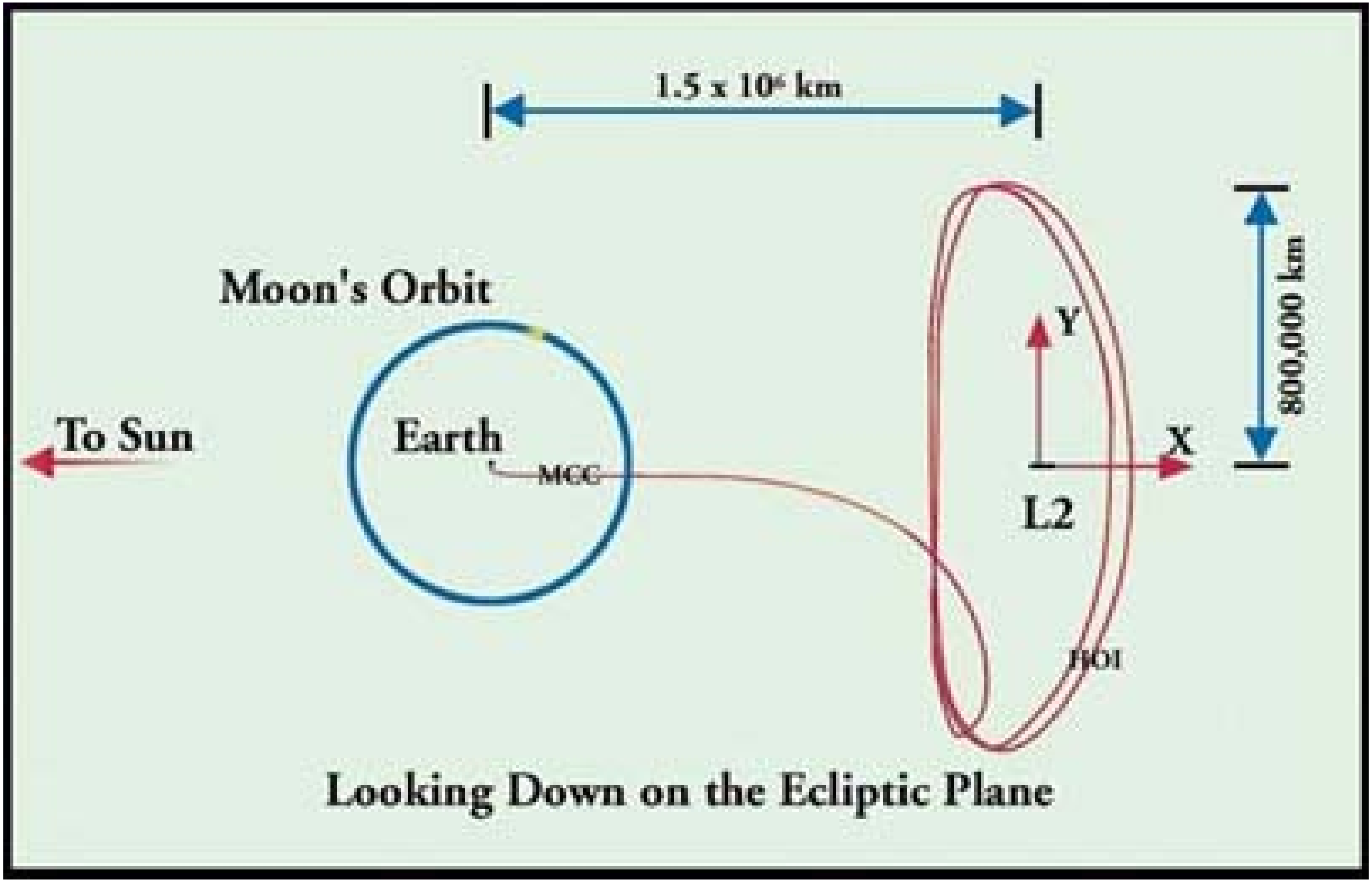
Sun-Earth Lagrange Points

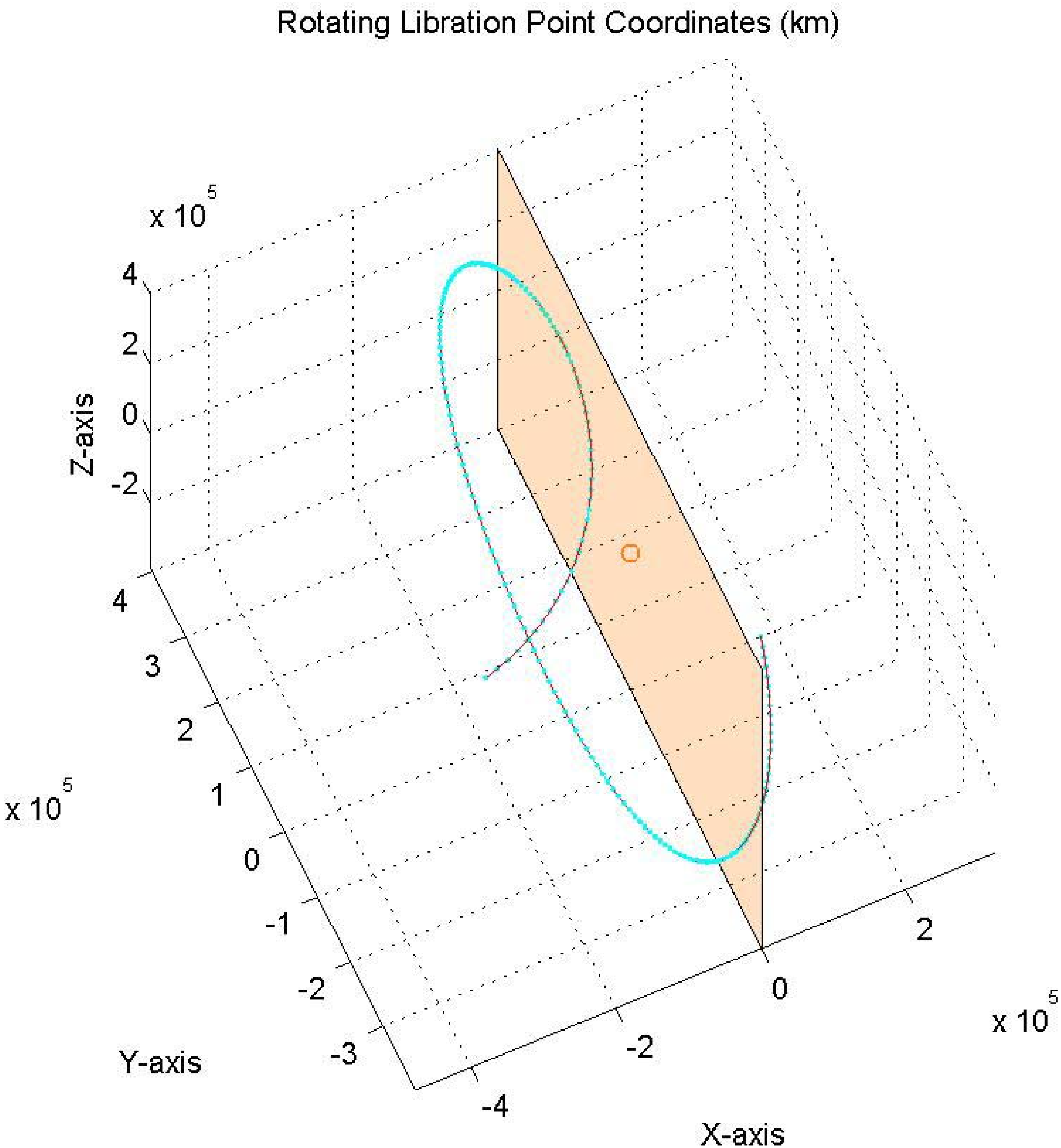


Equipotential Lines in Rotating Coordinates

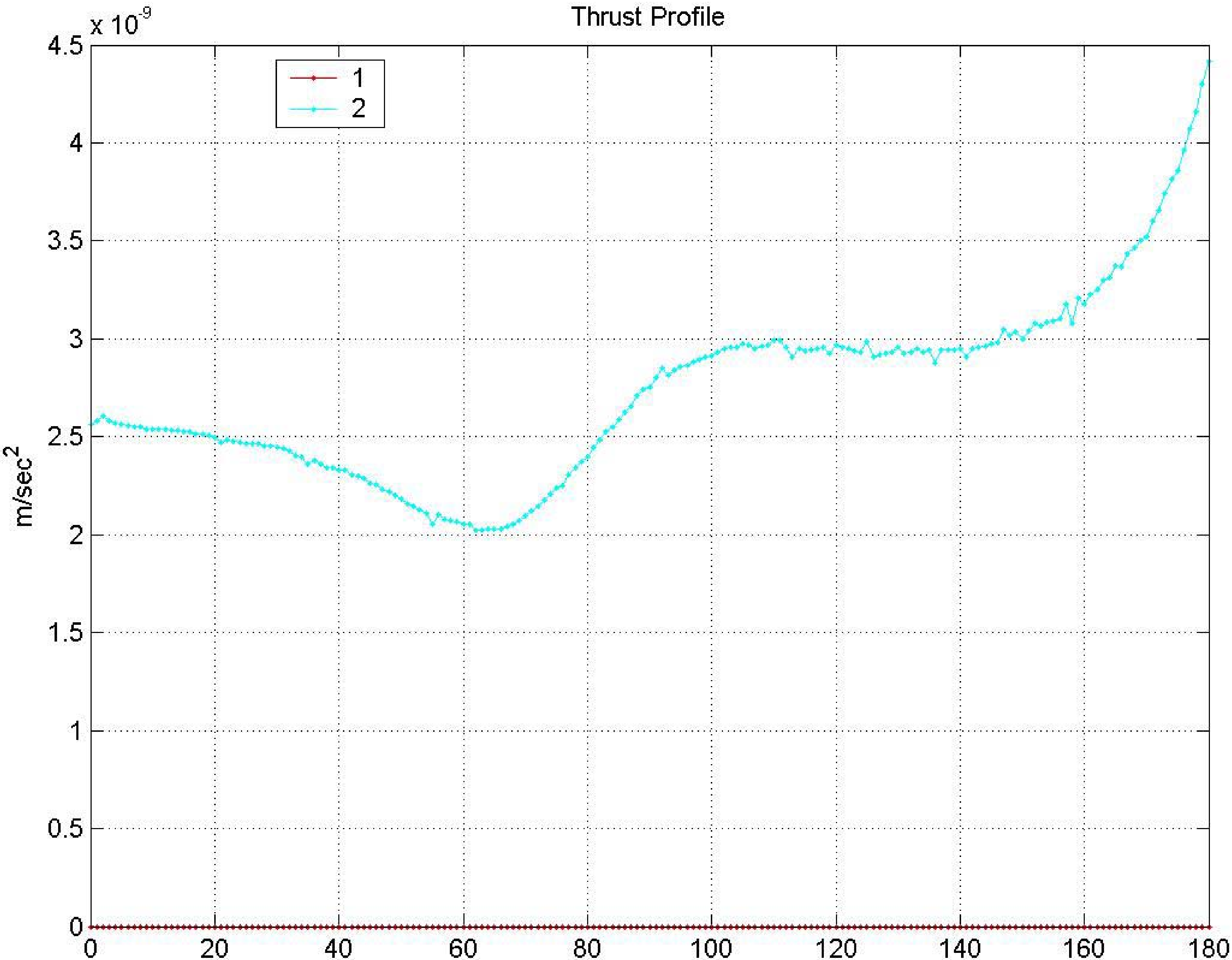


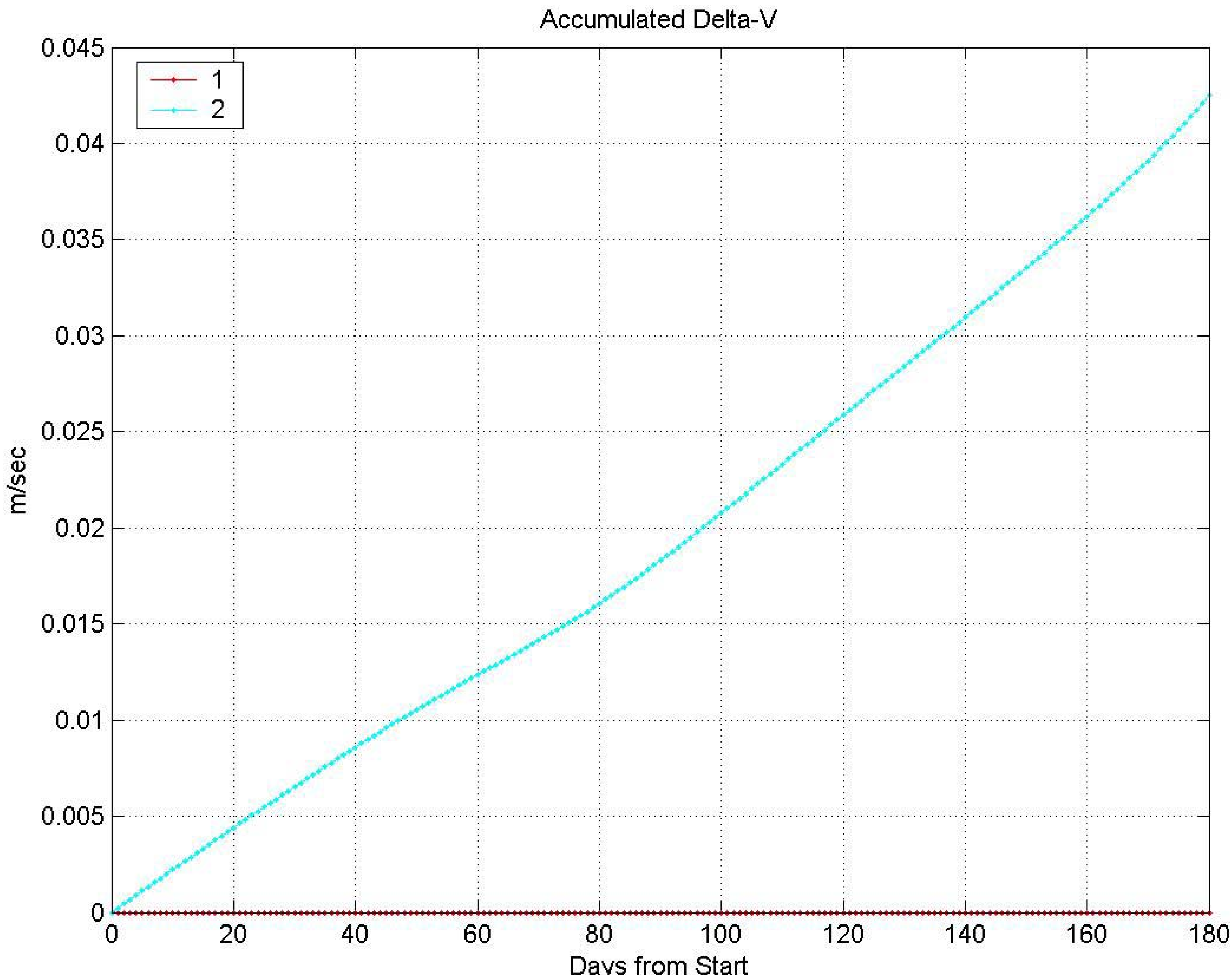
Halo Orbit About L2



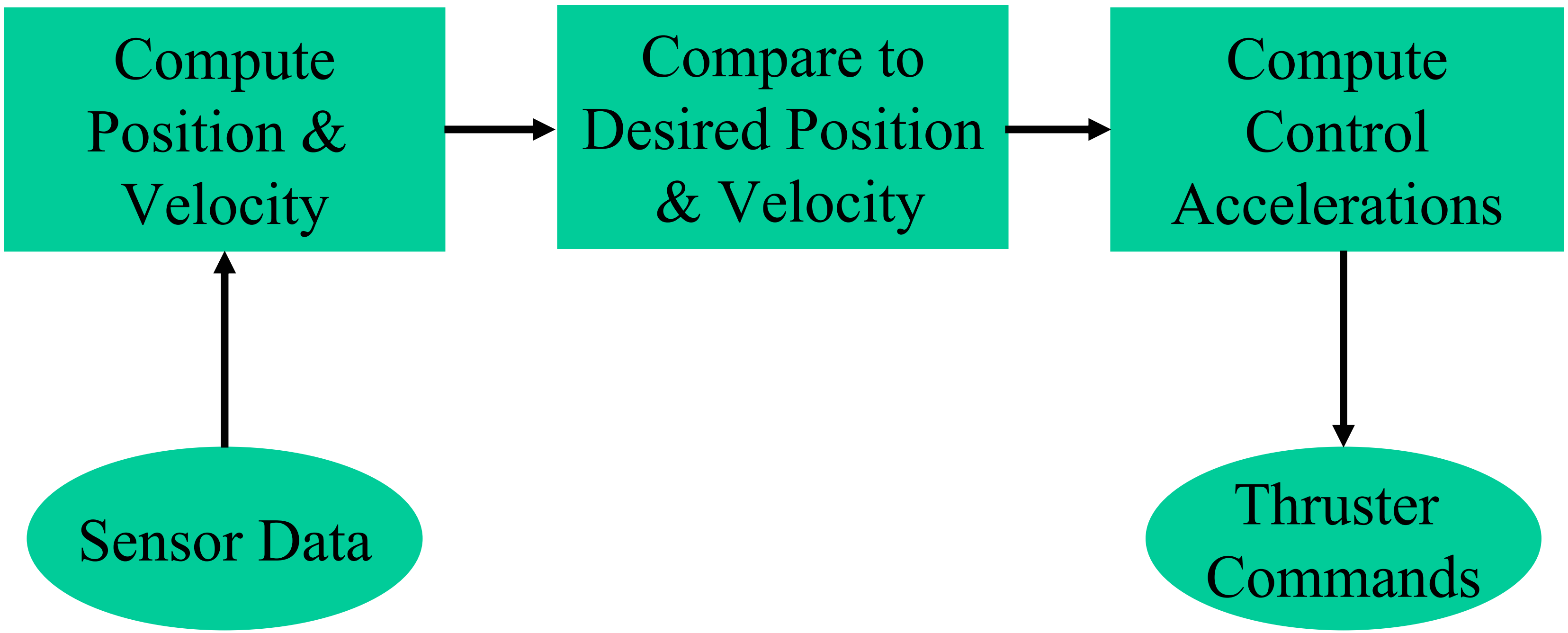


Rotating Libration Point Coordinates (km)
1
2
Z-axis
x 10^5
4
2
0
-2
Y-axis
x 10^5
4
3
2
1
0
-1
-2
-3
X-axis
x 10^5
-4
-2
0
2

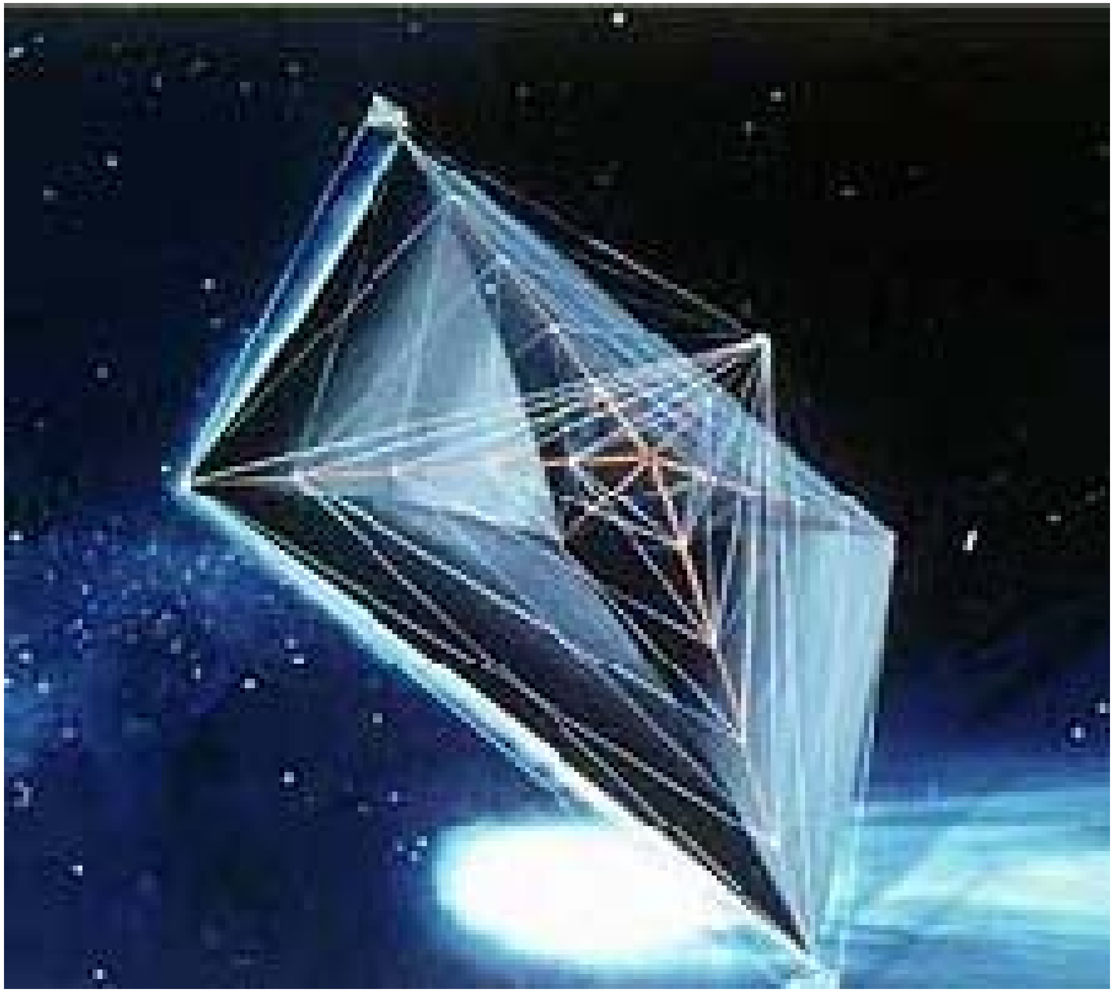

Thrust Profile
x 10^-9
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
m/sec^2
0
20
40
60
80
100
120
140
160
180
1
2

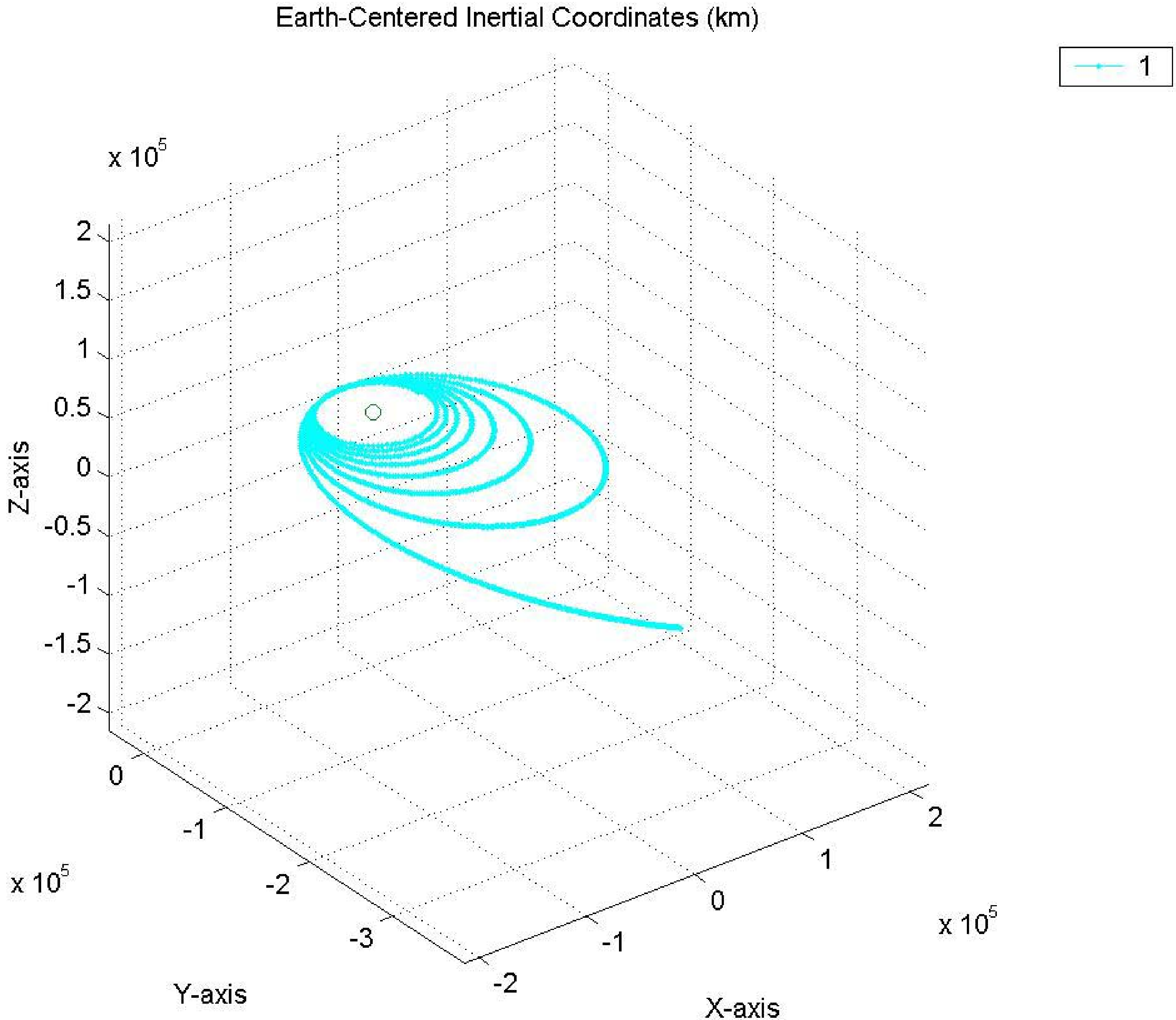

Accumulated Delta-V
1
2
m/sec
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
20
40
60
80
100
120
140
160
180
Days from Start

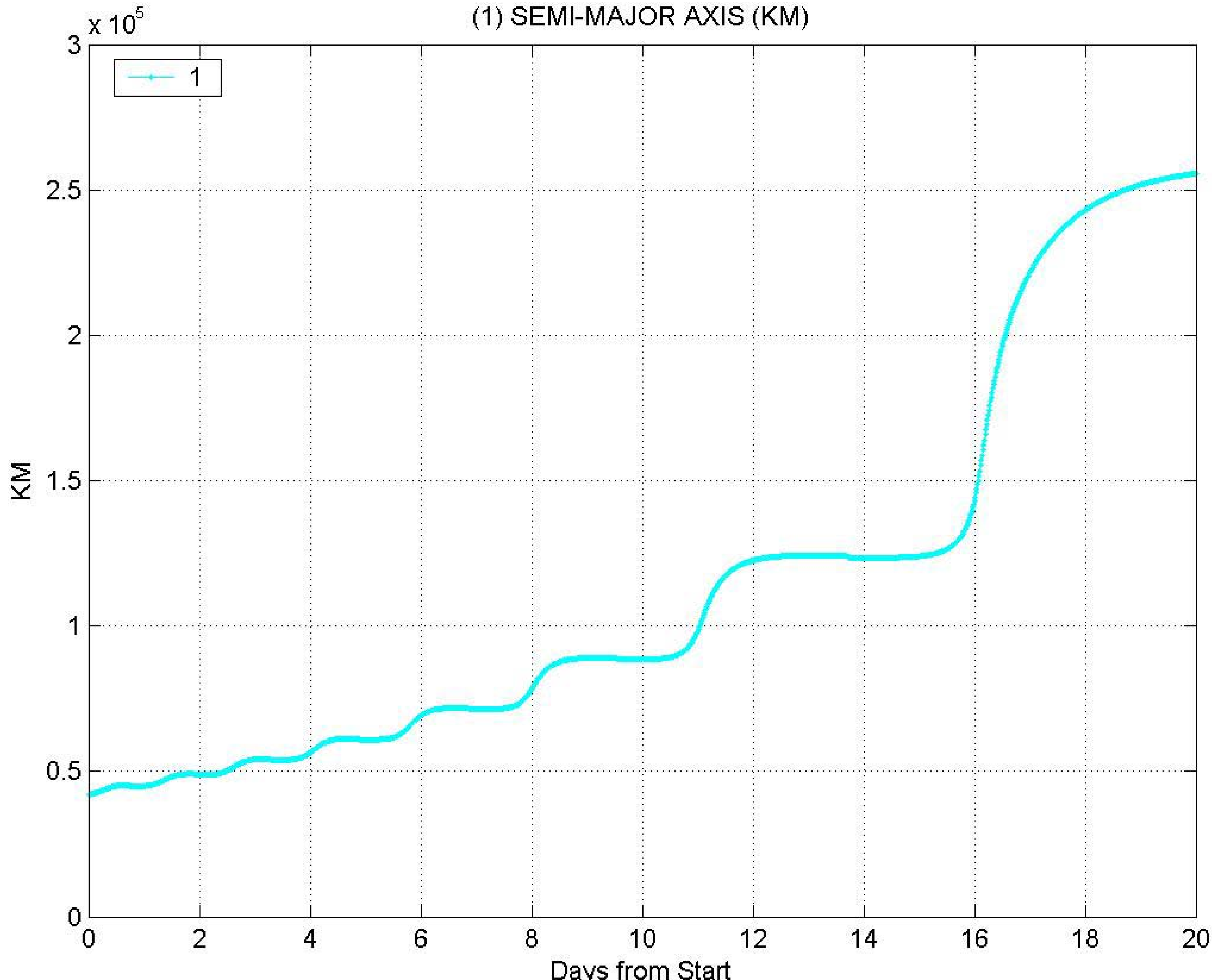
Control Law

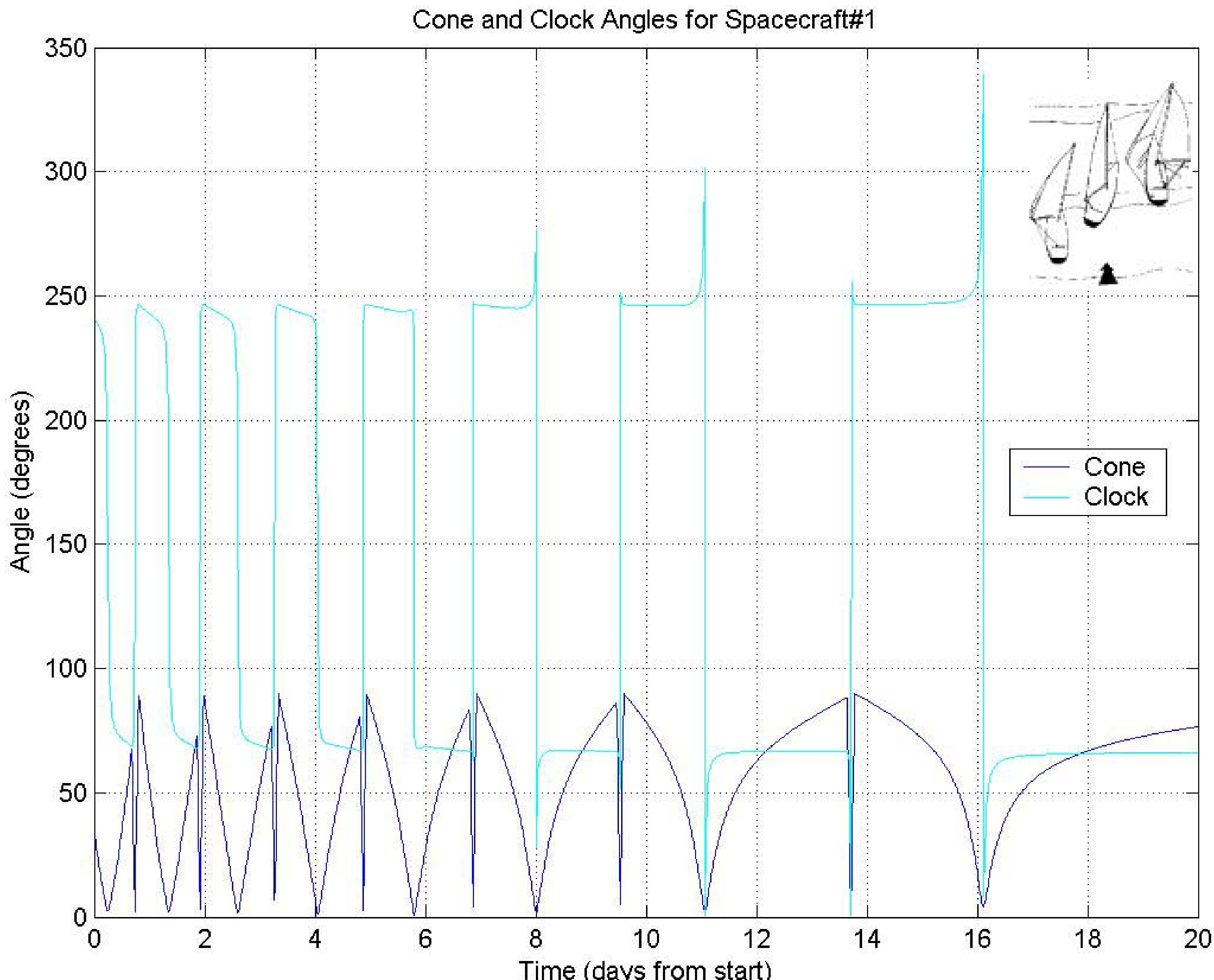

Compute Position & Velocity
Compare to Desired Position & Velocity
Compute Control Accelerations
Sensor Data
Thruster Commands

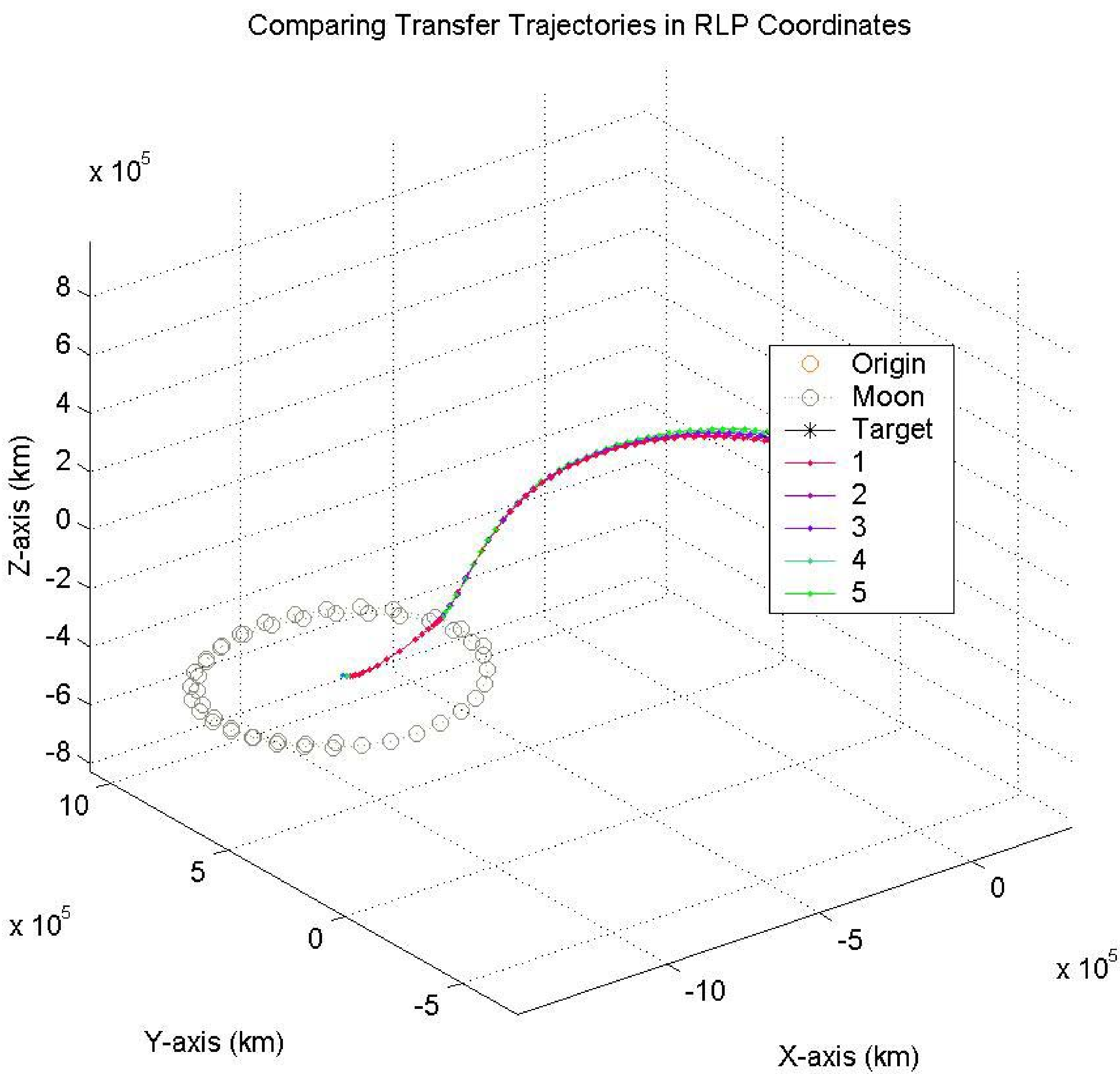
Solar Sail-Craft




Earth-Centered Inertial Coordinates (km)
1
x 10^5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
Z-axis
0
-1
-2
-3
x 10^5
Y-axis
-2
-1
0
1
2
x 10^5
X-axis


(1) SEMI-MAJOR AXIS (KM)
x 10^5
3
2.5
2
1.5
1
0.5
0
KM
1
0
2
4
6
8
10
12
14
16
18
20
Days from Start


Cone and Clock Angles for Spacecraft#1
Angle (degrees)
Time (days from start)
Cone
Clock
0
50
100
150
200
250
300
350
0
2
4
6
8
10
12
14
16
18
20


Comparing Transfer Trajectories in RLP Coordinates
x 10^5
8
6
4
2
0
-2
-4
-6
-8
Z-axis (km)
Origin
Moon
Target
1
2
3
4
5
10
5
0
-5
x 10^5
Y-axis (km)
0
-5
-10
x 10^5
X-axis (km)

Defining the Initial-Value Problem

Y = Positions and Velocities of Spacecraft

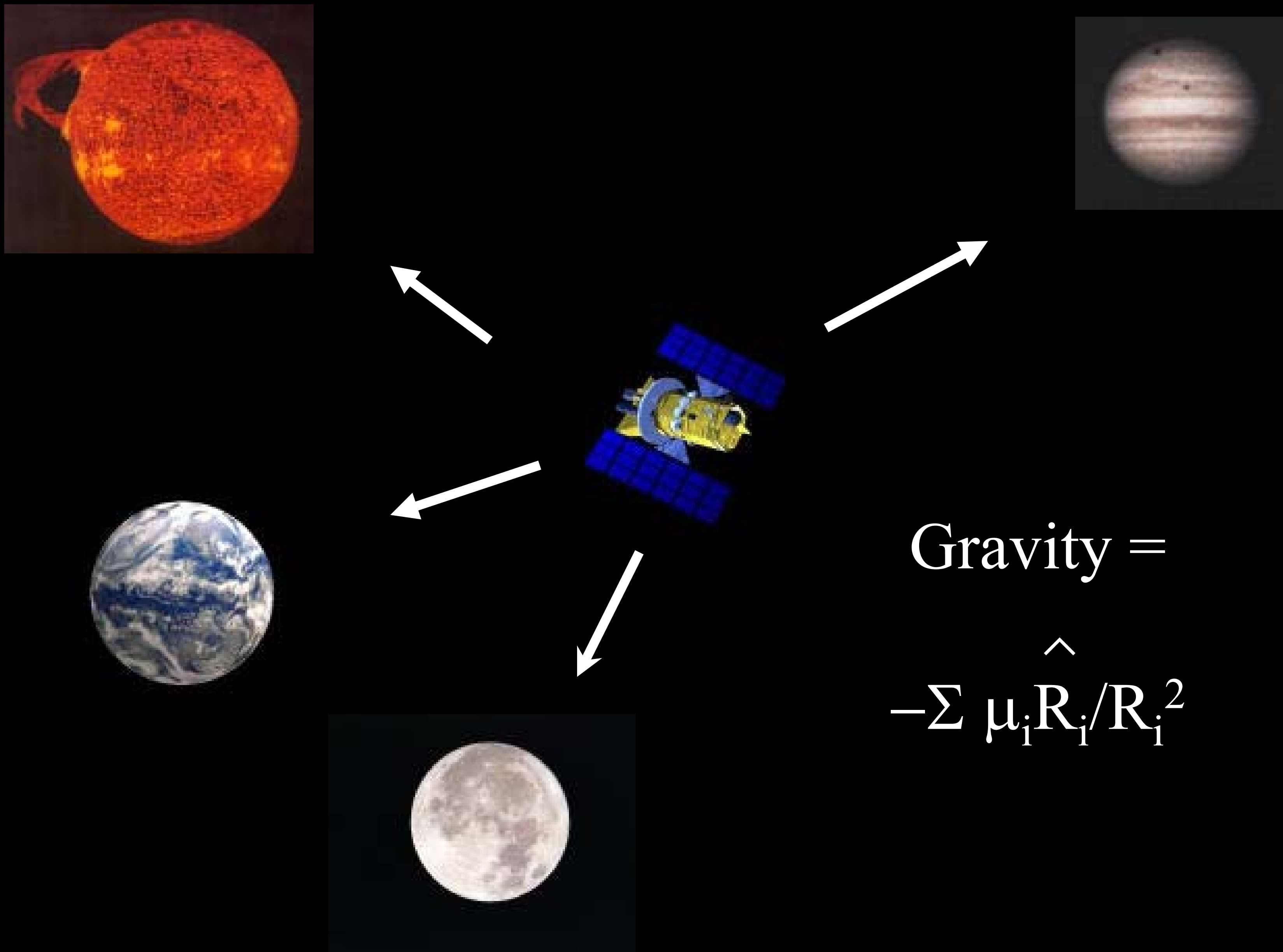
Y' = State Derivative

$$Y' = F(t, Y)$$

$$F = F_{gravity} + F_{env} + F_{control}$$

$$Y(t_0) = Y_0$$

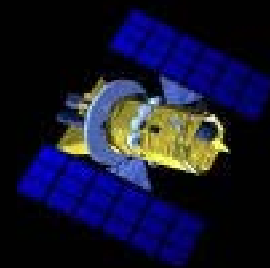
$$Y(t_0 \ldots t_f) = Solver(F, [t_0 \ldots t_f], Y_0, Options)$$



Gravity =

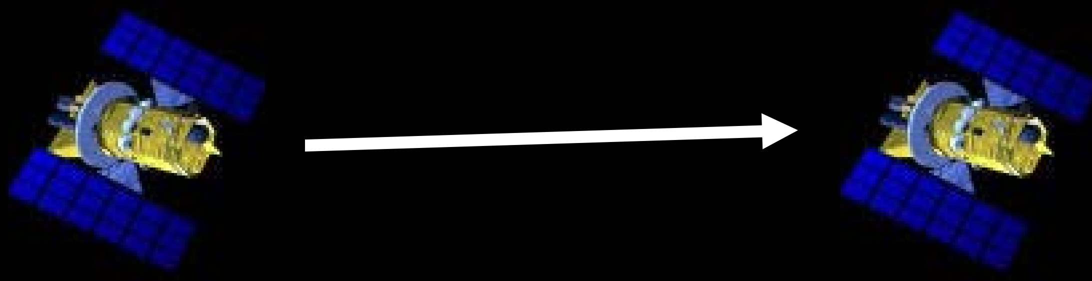
$$-\sum \mu_i \hat{R}_i / R_i^2$$

$$M_{sc}\vec{a} = (kS_0AD_0^2/c)\cos^2(\beta)\vec{R}/R^3$$



$$\vec{a} = -(1/2)\rho A C_D \vec{V}_{rel}/M_{sc}$$

Environmental Forces



$$\vec{a} = \vec{a}_{ref} - 2\Delta\vec{v}_{err}/\tau - 2\Delta\vec{r}_{err} / \tau^2$$

Control Force

Status

- 190K lines of code in configured library
- Demonstrated rapid implementation of new scenarios
- Implemented scripts to help users generate mex files
- Prepared PFFIAT Software Overview (version 1)

Major Challenge: Slow Execution!

CAUSES:

- Many spacecraft
- Expensive to compute the derivative
- High accuracy (10^{-12})
- Long time span (6 months to 1 year!)

Speeding Up Execution

- Choose the best solver (e.g., ODE113)
- Avoid "stiff" control laws
- Use PROFILE to identify "time hogs"
- MCC – Convert to C and compile
- Re-code in C or FORTRAN

Expected Future Work

- New Scenarios
  - Solar sail-craft transfer trajectory
  - Multiple spacecraft arrayed near L2
- New Control Laws
- New Environmental Models
  - Solar radiation
  - Air density and wind speed